

O Modelo e a Álgebra Relacional

O Modelo Relacional

- ▶ Modelo de dados, que se baseia no princípio em que todos os dados estão guardados em **tabelas**
- ▶ Baseado em lógica de predicados e na teoria de conjuntos.
- ▶ Sucessor do modelo hierárquico e do modelo em rede.

Modelo Relacional

- ▶ Um modelo relacional representa o **banco de dados** como um conjunto de **relações**.
- ▶ Uma **relação** pode ser pensada como uma **tabela de valores, ou lista de tuplas**.
- ▶ Cada linha na tabela representa uma coleção de valores de dados relacionados.

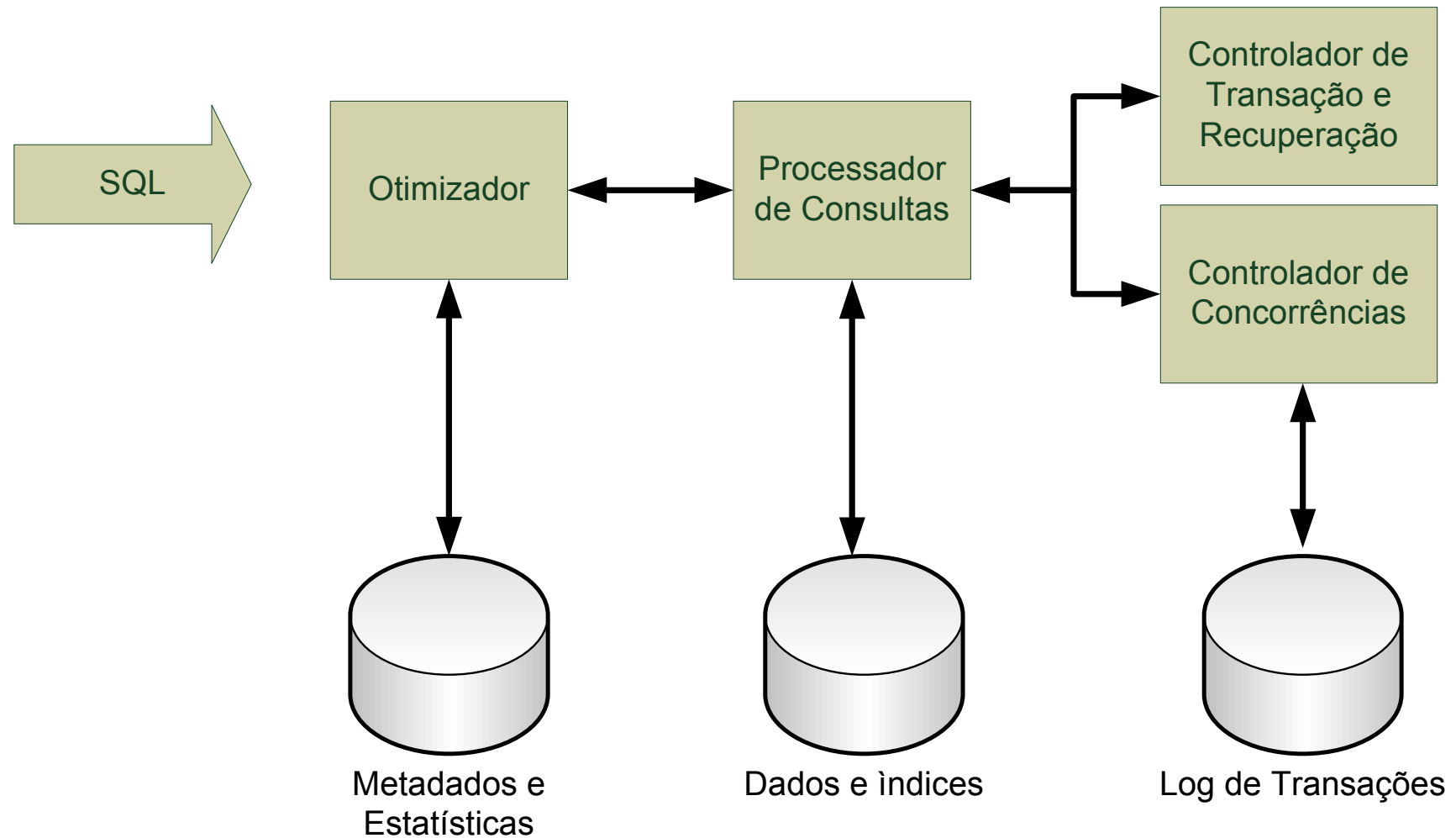
O Modelo Relacional

- ▶ Primeiro modelo de banco de dados formal
 - ▶ Somente depois seus antecessores, os bancos de dados hierárquicos e em rede, passaram a ser também descritos em linguagem formal.
- ▶ A linguagem padrão para os bancos de dados relacionais é o SQL, (*structured query language*)
 - ▶ DDL e DML

O Modelo Relacional

- ▶ A principal proposição do modelo relacional é que todos os dados são representados como **relações** matemáticas
- ▶ um subconjunto do produto Cartesiano de n conjuntos.
- ▶ No modelo matemático (diferentemente do SQL), a análise dos dados é feita em uma lógica de predicados de dois valores (ou seja, sem o valor nulo);
 - ▶ Existem apenas dois possíveis valores para uma proposição: verdadeira ou falsa.
- ▶ Os dados são tratados pelo cálculo relacional ou álgebra relacional.

Arquitetura Funcional de um SGBD Relacional



Arquitetura Funcional

- ▶ A interação principal é através de comandos SQL
- ▶ Armazena-se não somente dados, mas também índices, Logs, metadados
- ▶ Processadores controladores internos fazem parte de um sistema chamado “Subsistema de armazenamento”

Otimizador/ Processador de Consultas

- ▶ Tratamento da requisição da linguagem de alto nível, verificando a sintaxe e semântica com relação ao esquema estabelecido
- ▶ Otimização (heurística) de consultas
- ▶ Determinação do plano de acesso aos dados, indicando estruturas de armazenamentos e índices

Controlador de Transações, Concorrências e Recuperação

- ▶ Controle da unidade básica do SGBD – transações
 - ▶ Tudo para o SGBD é uma transação
 - ▶ Uma consulta simples é uma transação
 - ▶ Uma atualização numa tupla – é transação
 - ▶ Transações implementam requisições de vários usuários
- ▶ Garante a consistência de dados com execução concorrente de múltipla transações
 - ▶ SGBD simula multiprocessamento
- ▶ Gerencia de bloqueios
- ▶ Responsável pelo Log do sistema e recuperação

Metabase

- ▶ **Base de dados da base de Dados!!!**
 - ▶ Contem informações sobre os BDs armazenados e sobre o ambiente
 - ▶ Utilizada pelo DBA para administração dos BDS e gerencia do SGBD
- ▶ Também conhecida como dicionário de dados ou catálogo

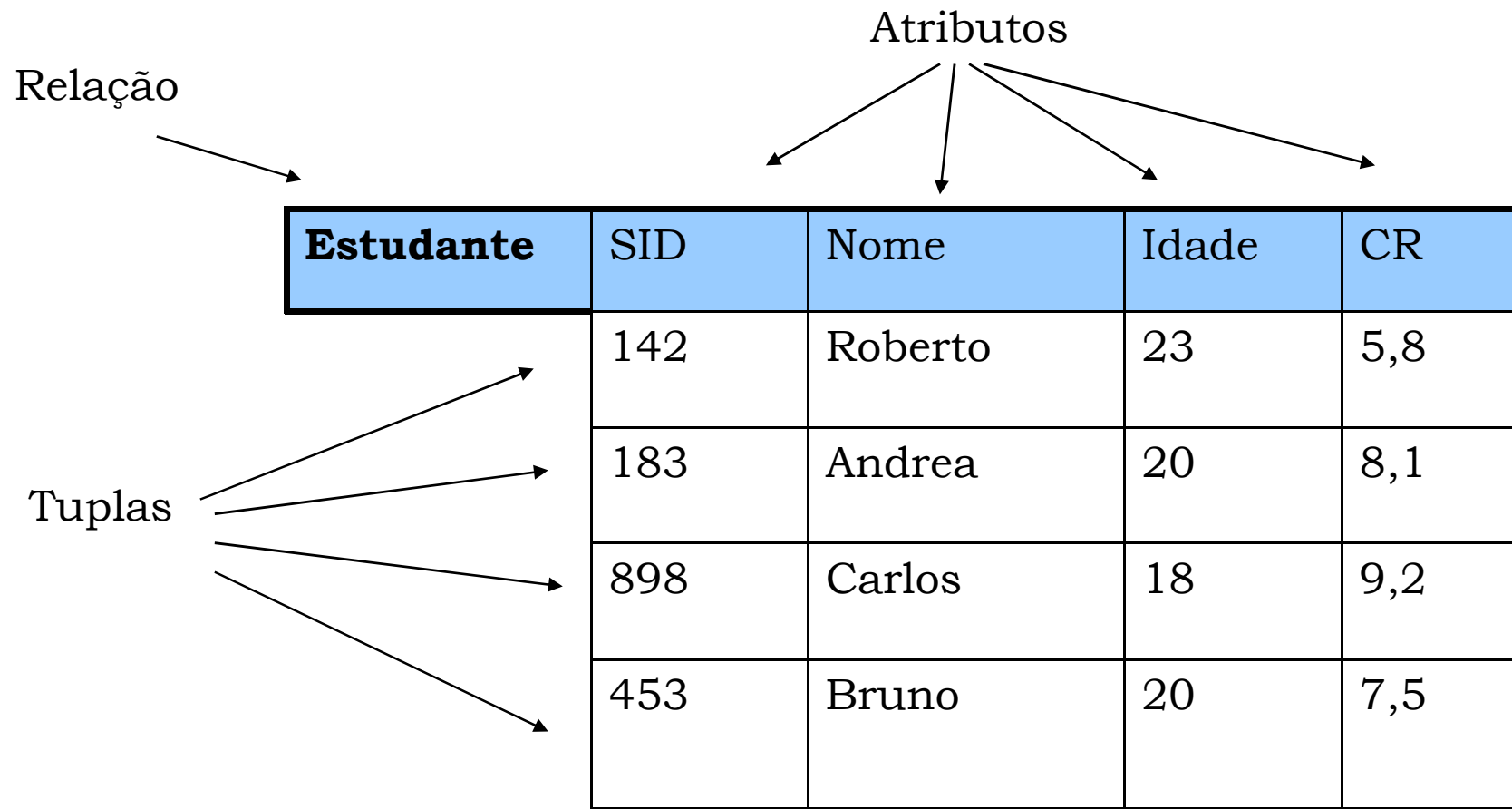
Manipulação de dados Relacionais

- ▶ Duas categorias de linguagens
 - ▶ formais
 - ▶ Álgebra Relacional e Cálculo Relacional
 - ▶ comerciais (baseadas nas linguagens formais)
 - ▶ SQL
- ▶ Linguagens formais - Características
 - ▶ Orientadas a conjuntos
 - ▶ Linguagens de base
 - ▶ linguagens relacionais devem ter no mínimo um poder de expressão equivalente ao de uma linguagem formal
 - ▶ fechamento
 - ▶ resultados de consultas são relações

Definido o Modelo Relacional

- ▶ Formalmente, na terminologia formal de modelo relacional, temos as seguintes definições:
 - ▶ **Tupla**: uma linha
 - ▶ **Atributo**: é o cabeçalho da coluna
 - ▶ **Relação**: é a tabela
 - ▶ **Domínio**: são os valores aceitáveis para um atributo
- ▶ Exemplo: Seja o esquema
- ▶ Aluno(Nome, Mat, FoneResidencia, Endereco, FoneComercial, Idade, CR)
 - ▶ **Relação**: Aluno
 - ▶ **Atributo**: (7 atributos) - Aluno(Nome, Mat, FoneResidencia, Endereco, FoneComercial, Idade, CR)
 - ▶ **Domínio**: Dom(Nome)=nomes; Dom(Mat)= números
 - ▶ **Tupla**: <Roberto, 001,222-2222,rua ..., 98765432, 40, 9,1>

Definido o Modelo Relacional Exemplo



Características das Relações

► Ordenação de tuplas numa relação

- **Relação é um conjunto de tuplas**; matematicamente falando os elementos de um conjunto não tem ordem, portanto **uma relação não tem qualquer ordem**
- Entretanto como trata-se de dados num computador sempre há uma ordem de armazenamento físico no disco

► Unicidade

- Matematicamente, num conjunto não existem elementos duplicados
- Assim, **não existem tuplas idênticas**

Restrições do Modelo Relacional

- ▶ Limitações nos dados
- ▶ Três grupos:
 - ▶ Restrições Inerentes ao modelo de Dados
 - ▶ Restrições baseadas no esquema
 - ▶ Expressas no modelo
 - ▶ Restrições baseadas nas aplicações
 - ▶ Impostas pela aplicação

Atributos Chaves

▶ Superchave

- ▶ Conjunto de um ou mais atributos que nos permite identificar inequivocamente uma entidade em um conjunto de entidades.
- ▶ Duas tuplas não podem ter a mesma superchave
- ▶ A superchave pode conter atributos desnecessários.
 - ▶ Por exemplo: A relação Pessoa pode ter como superchave
 - Pessoa(CPF, Nome, endereço,)
 - Superchaves possíveis
 - {CPF}
 - {CPF, nome}
 - {CPF, nome, Endereço}
 - Etc...

Atributos Chaves

- ▶ **Chave**

- ▶ Superchave mínima
- ▶ Pode existir mais de uma chave nesta condição: Chave Candidata

- ▶ **Chave Candidata**

- ▶ são as menores superchaves possíveis (cujos subconjuntos não sejam superchaves),

EMPREGADO(ENOME, CPF, MAT, NDATA, ENDERECO,
DEPNUM*)

Chaves.

{CPF} e {MAT}

- ▶ **Chave Primária** -> Chave Candidata escolhida

- ▶ Por convenção, ela aparece sublinhada num esquema

Chave Estrangeira

- ▶ É necessário manter a consistência entre **tuplas** de duas **relações**: uma **tupla** em uma **relação** que se refere a uma outra **relação** deve referenciar uma **tupla** existente naquela relação.
- ▶ O **atributo** (ou **conjunto de atributos**) de uma **relação** R1 que referencia uma outra relação R2 é chamado de **chave estrangeira**.
 - ▶ Atenção, pois o **atributo não é chave de R1**,
- ▶ O valor da chave estrangeira em uma **tupla** de R1 deve existir como o valor da **chave primária** de alguma **tupla** de R2
 - ▶ Ou, se não for obrigatório, pode ser nulo.

Exemplo de Chaves Primária e Estrangeira

Departamento(Cod_Departamento, Departamento, ...)

Funcionario(Cod_Funcionario, Nome, *Cod_Departamento*, ...)

Departamentos

| Cod_Departamento | Departamento | ... |
|------------------|--------------|-----|
| D01 | Informática | |
| D02 | Matemática | |
| D03 | Gestão | |
| D04 | Direito | |

Chave Primária: Cod_Departamento

Funcionários

| Cod_Funcionario | Nome | Cod_Departamento | ... |
|-----------------|------------|------------------|-----|
| D01 | Bernardino | D01 | |
| D02 | Lucas | D04 | |
| D03 | Ernestina | D02 | |
| D04 | Lucrecia | D03 | |

Chave Primária: Cod_Funcionario

Chave Estrangeira: Cod_Departamento

Álgebra Relacional

▶ Básicas

- ▶ **Seleção (σ)** - seleciona um subconjunto de linhas de uma relação
- ▶ **Projeção (π)** – apaga colunas desnecessárias de uma relação
- ▶ **Produto cartesiano (\times)** – permite combinar duas relações
- ▶ **União (\cup)** - tuplas na relação 1 e na relação 2
- ▶ **Diferença ($-$)** – tuplas na relação 1 mas não na relação 2
- ▶ **Renomeação (ρ)** – renomeia tabela
- ▶ **Atribuição (\leftarrow)** – Atribui valores a variáveis

▶ Derivadas

- ▶ Junção, interseção, divisão
 - ▶ muito úteis!

Seleção

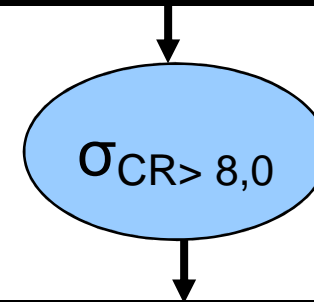
- ▶ **Notação:** σ_p
 - ▶ “p” é chamado predicado/condição de seleção
- ▶ **Entrada:** Tabela (R)
 - ▶ Operadores de comparação (=, <, <=, >, >=) e Operadores lógicos: \wedge (and), \vee (or) e \neg (not)
- ▶ **Propósito:** filtragem de linhas de acordo com um critério
- ▶ **Saída:** mesmas colunas de R, porém somente as linhas de R que satisfazem ao critério p

Seleção

Exemplo

- ▶ Estudantes com $CR > 8,0$
 - ▶ $\sigma_{CR > 8,0}$ (estudante)

| SID | Nome | Idade | CR |
|-----|---------|-------|-----|
| 142 | Roberto | 23 | 5,8 |
| 183 | Andrea | 20 | 8,1 |
| 898 | Carlos | 18 | 9,2 |
| 453 | Bruno | 20 | 7,5 |



| SID | Nome | Idade | CR |
|-----|-------------------|-------|-----|
| | | | |
| 183 | Andrea | 20 | 8,1 |
| 898 | Carlos | 18 | 9,2 |
| | | | |
| | Roberto Harkovsky | | |

Seleção

- ▶ Um predicado de seleção pode incluir, em geral, qualquer coluna de R, constantes, comparações ($=$, $<$, etc., $>$) e conectivos booleanos (\wedge , \vee , and \sim)
- ▶ Exemplo: Estudantes com $CR > 8,0$ e menores de 18 ou maiores que 21 anos
 - ▶ $\sigma_{cr > 8,0 \wedge (idade < 18 \vee idade > 21)}$ (estudante)

Projeção

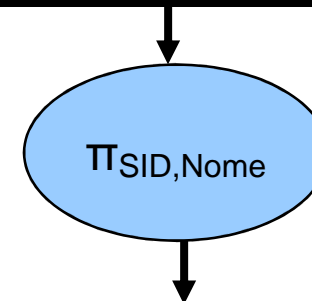
- ▶ **Notação:** π_L
 - ▶ “L” lista de colunas de R
- ▶ **Entrada:** Tabela (R)
- ▶ **Propósito:** seleciona colunas em R
- ▶ **Saída:** mesmas linhas de R, porém somente as colunas de R que satisfazem ao critério L

Projeção

Exemplo

- ▶ SID e nome dos estudantes
 - ▶ $\pi_{SID, Nome}(\text{estudante})$

| SID | Nome | Idade | CR |
|-----|---------|-------|-----|
| 142 | Roberto | 23 | 5,8 |
| 183 | Andrea | 20 | 8,1 |
| 898 | Carlos | 18 | 9,2 |
| 453 | Bruno | 20 | 7,5 |



| SID | Nome | | |
|-------------------|---------|--|--|
| 142 | Roberto | | |
| 183 | Andrea | | |
| 898 | Carlos | | |
| 453 | Bruno | | |
| Roberto Harkovsky | | | |

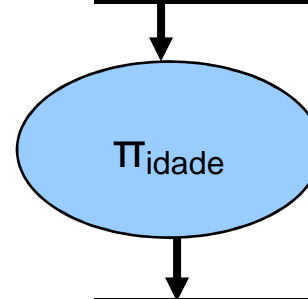
Exemplo2

- ▶ SID e nome dos estudantes

- ▶ $\pi_{\text{idade}}(\text{estudante})$

- ▶ Duplicidade deve ser eliminada

| SID | Nome | Idade | CR |
|-----|---------|-------|-----|
| 142 | Roberto | 23 | 5,8 |
| 183 | Andrea | 20 | 8,1 |
| 898 | Carlos | 18 | 9,2 |
| 453 | Bruno | 20 | 7,5 |



| | | | |
|--|--|-------|--|
| | | Idade | |
| | | 23 | |
| | | 20 | |
| | | 18 | |
| | | | |

Produto Cartesiano

- ▶ **Notação:** $R \times S$
- ▶ **Entrada:** Tabela (R) e Tabela (S)
- ▶ **Propósito:** gera combinações de linhas das duas tabelas
- ▶ **Saída:** Para cada linha r em R e cada linha s em S , gerar a tupla rs

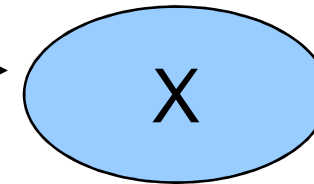
Produto Cartesiano

Exemplo

► Estudante X Matricula

| SID | Nome | Idade | CR |
|-----|---------|-------|-----|
| 142 | Roberto | 23 | 5,8 |
| 183 | Andrea | 20 | 8,1 |
| 898 | Carlos | 18 | 9,2 |
| 453 | Bruno | 20 | 7,5 |

| SID | CID |
|-----|--------|
| 142 | CPS116 |
| 183 | CPS114 |
| 898 | CPS116 |



| SID | Nome | Idade | CR | SID | CID |
|-------|---------|-------|-------|-------|--------|
| 142 | Roberto | 23 | 5,8 | 142 | CPS116 |
| 142 | Roberto | 23 | 5,8 | 183 | CPS114 |
| 142 | Roberto | 23 | 5,8 | 898 | CPS116 |
| 183 | Andrea | 20 | 8,1 | 142 | CPS116 |
| | | | | | |

Nota sobre Produto Cartesiano

- ▶ A ordem das colunas não é importante
 - ▶ Assim a operação é comutativa
 - ▶ $R \times S = S \times R$

União

- ▶ **Notação:** $R \cup S$
 - ▶ R e S devem ter o mesmo esquema
- ▶ **Entrada:** Tabela (R) e Tabela (S)
- ▶ **Propósito:** gera linhas de acordo com um critério
- ▶ **Saída:** Contém todas as linhas de R e de S
 - ▶ O esquema é o mesmo das tabelas de entrada
 - ▶ Duplicidade é eliminada

Renomeação

- ▶ **Notação:** $\rho_S(R)$ ou $\rho_{S(A_1, A_2, \dots)}(R)$
 - ▶ Renomeia R para S ou renomeia R para S com atributos renomeados A1, A2, ...
- ▶ **Entrada:** Tabela (R)
- ▶ **Propósito:** redefinir nome tabelas / ou colunas num contexto
- ▶ **Saída:** Tabela renomeada com mesmas linhas de R
- ▶ Usada para
 - ▶ Útil para auto-relacionamentos, onde precisamos fazer a junção de uma tabela com ela mesma, e nesse caso cada versão da tabela precisa receber um nome diferente da outra.
 - ▶ Cria colunas idênticas numa junção natural

Junção

- ▶ **Notação:** $R \bowtie_p S$
 - ▶ “p” é a condição/predicado da junção
- ▶ **Entrada:** Tabela (R) e Tabela (S)
- ▶ **Propósito:** gera linhas de acordo com um critério
- ▶ **Saída:** Para cada linha r em R e cada linha sem S, gerar a tupla rs se, e somente se, atenderem a condição p
- ▶ Abreviação para $\sigma_p (R \times S)$

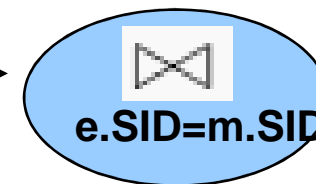
Exemplo

Estudante e

| SID | Nome | Idade | CR |
|-----|---------|-------|-----|
| 142 | Roberto | 23 | 5,8 |
| 183 | Andrea | 20 | 8,1 |
| 898 | Carlos | 18 | 9,2 |
| 453 | Bruno | 20 | 7,5 |

Matricula m

| SID | CID |
|-----|--------|
| 142 | CPS114 |
| 142 | CPS116 |
| 183 | CPS114 |
| 898 | CPS116 |



| SID | Nome | Idade | CR | SID | CID |
|-----|---------|-------|-----|-----|--------|
| 142 | Roberto | 23 | 5,8 | 142 | CPS114 |
| 142 | Roberto | 23 | 5,8 | 142 | CPS116 |
| 183 | Andrea | 20 | 8,1 | 183 | CPS114 |
| 898 | Carlos | 18 | 9,2 | 898 | CPS116 |

Operador Derivado

Junção Natural

- ▶ **Notação:** $R \bowtie S$
- ▶ **Entrada:** Tabela (R) e Tabela (S)
- ▶ **Propósito:** relaciona linhas das tabelas
 - ▶ Reforça a igualdade de seus atributos
 - ▶ Elimina 1 cópia dos atributos comuns
- ▶ **Saída:** Para cada linha r em R e cada linha s em S , gerar a tupla rs se, e somente se, atenderem a condição p

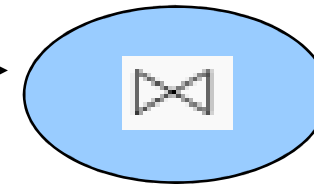
Exemplo

Estudante e

| SID | Nome | Idade | CR |
|-----|---------|-------|-----|
| 142 | Roberto | 23 | 5,8 |
| 183 | Andrea | 20 | 8,1 |
| 898 | Carlos | 18 | 9,2 |
| 453 | Bruno | 20 | 7,5 |

Matricula m

| SID | CID |
|-----|--------|
| 142 | CPS114 |
| 142 | CPS116 |
| 183 | CPS114 |
| 898 | CPS116 |



| SID | Nome | Idade | CR | | CID |
|-----|---------|-------|-----|--|--------|
| 142 | Roberto | 23 | 5,8 | | CPS114 |
| 142 | Roberto | 23 | 5,8 | | CPS116 |
| 183 | Andrea | 20 | 8,1 | | CPS114 |
| 898 | Carlos | 18 | 9,2 | | CPS116 |

Diferença

- ▶ **Notação:** R - S
 - ▶ R e S devem ter o mesmo esquema
- ▶ **Entrada:** Tabela (R) e Tabela (S)
- ▶ **Propósito:** gera linhas de acordo com um critério
- ▶ **Saída:** Contém todas as linhas de R e que não são encontradas em S
 - ▶ O esquema é o mesmo das tabelas de entrada

Operador Derivado - Interseção

- ▶ **Notação:** $R \cap S$
 - ▶ R e S devem ter o mesmo esquema
- ▶ **Entrada:** Tabela (R) e Tabela (S)
- ▶ **Propósito:** gera linhas de acordo com um critério
- ▶ **Saída:** Contém todas as linhas de R que são encontradas em S também
 - ▶ O esquema é o mesmo das tabelas de entrada
 - ▶ $R - (R - S)$ ou $S - (S - R)$ ou $R \cap S$



Divisão

- ▶ **Notação:** $R \div S$
- ▶ **Entrada:** Tabela (R) e Tabela (S)
 - ▶ Seja grau a medida de atributos de mesmo nome
 - ▶ R tem grau (“m”+”n”)
 - ▶ S tem grau “n”
- ▶ **Propósito:** gera linhas de acordo com um critério
- ▶ **Saída:** atributos de S cujos valores associam-se com todos os valores de R
 - ▶ Grau “m”

Divisão

Exemplo

$$R_1$$

| x | y | z |
|-----|-----|-----|
| 1 | 1 | 1 |
| 1 | 2 | 1 |
| 2 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 1 | 3 |

$$R_2a$$

| z |
|-----|
| 1 |

$$R_2b$$

| y | z |
|-----|-----|
| 1 | 1 |

$$R_2c$$

| y |
|-----|
| 1 |
| 2 |

$$R_1 \stackrel{\cdot}{\div} R_2a$$

| x | y |
|-----|-----|
| 1 | 1 |
| 1 | 2 |
| 2 | 1 |

$$R_1 \stackrel{\cdot}{\div} R_2b$$

| x |
|-----|
| 1 |
| 2 |

$$R_1 \stackrel{\cdot}{\div} R_2c$$

| x | z |
|-----|-----|
| 1 | 1 |

Atribuição

- ▶ Armazena o resultado de uma expressão algébrica em uma variável de relação
 - ▶ permite o processamento de uma consulta complexa em etapas
- ▶ Notação
 - ▶ *nomeVariável* \leftarrow *expressãoÁlgebra*
- ▶ Exemplo
 - ▶ $r1 \leftarrow \sigma_{\text{nome}='bob'}(\text{estudante})$

Funções Agregadas

- ▶ Para aquelas consultas que não podem ser resolvidas simplesmente através da álgebra relacional, introduz-se um conjunto de funções agregadas
- ▶ Funções comumente aplicadas a conjuntos de dados são: Média, Máximo, Mínimo, Soma, Contador

Resumo

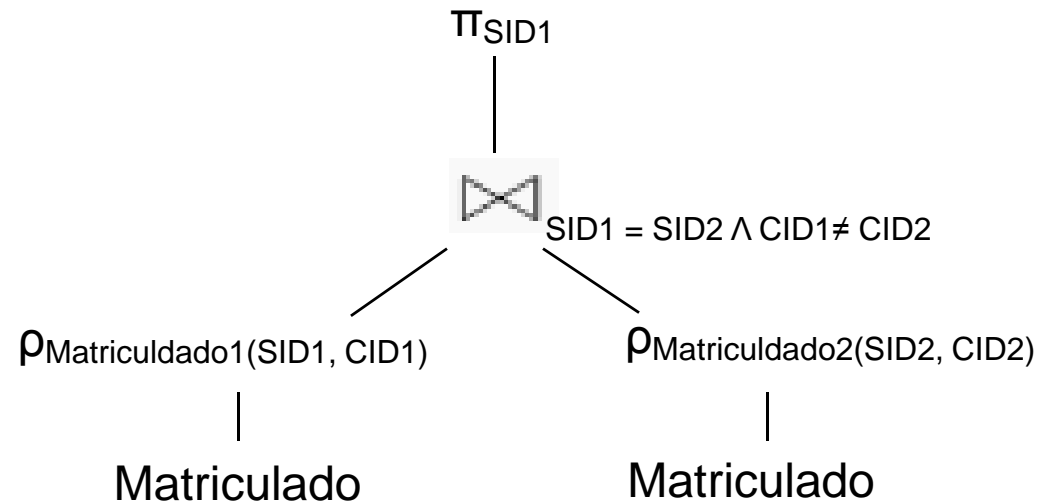
| <i>Símbolo</i> | <i>Operação</i> | <i>Sintaxe</i> | <i>Tipo</i> |
|----------------|------------------------|---|-------------|
| σ | Seleção / Restrição | $\sigma_{\text{condição}} (\text{Relação})$ | Primitiva |
| π | Projeção | $\pi_{\text{expressões}} (\text{Relação})$ | Primitiva |
| \cup | União | $\text{Relação1} \cup \text{Relação2}$ | Primitiva |
| \cap | Intersecção | $\text{Relação1} \cap \text{Relação2}$ | Adicional |
| - | Diferença de conjuntos | $\text{Relação1} - \text{Relação2}$ | Primitiva |
| x | Produto cartesiano | $\text{Relação1} \times \text{Relação2}$ | Primitiva |
| x | Junção | $\text{Relação1} \bowtie \text{Relação2}$ | Adicional |
| \div | Divisão | $\text{Relação1} \div \text{Relação2}$ | Adicional |
| ρ | Renomeação | $\rho_{\text{nome}} (\text{Relação})$ | Primitiva |
| \leftarrow | Atribuição | $\text{variável} \leftarrow \text{Relação}$ | Adicional |

Exemplos

- ▶ *Para os exemplos a seguir considere o seguinte esquema:*
 - ▶ Estud (SID, nome, idade, CR)
 - ▶ Curso (CID, titulo)
 - ▶ Matr (SID, CID)

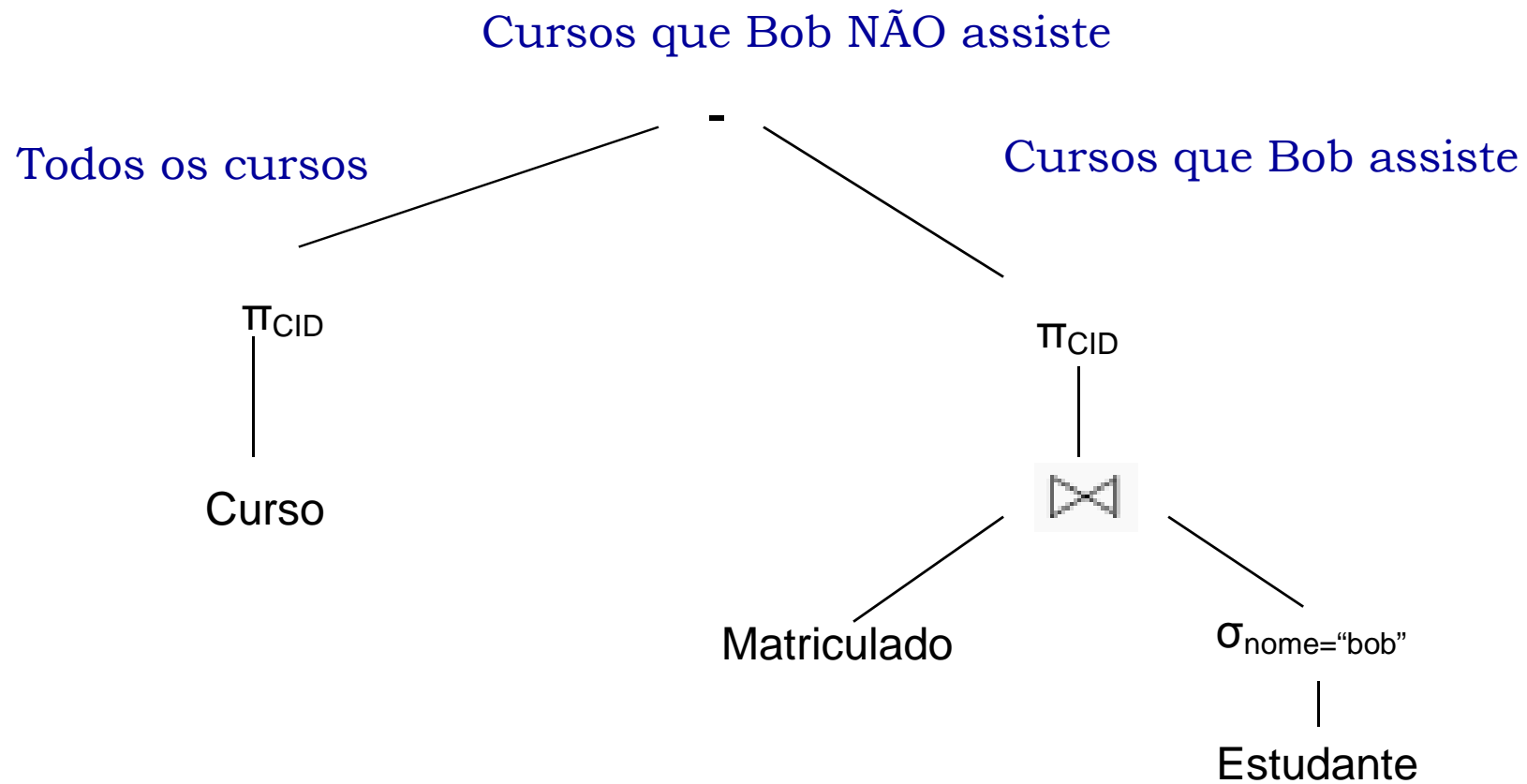
Exemplo

- ▶ SID dos estudantes que assistem a pelo menos 2 cursos
- ▶ $r1 \leftarrow \text{SELECIONE Matric COM SID=SID1 E CID} = \text{CID1}$
- ▶ $r2 \leftarrow \text{SELECIONE Matric COM SID=SID2 E CID} = \text{CID2}$
- ▶ $r3 \leftarrow \text{JUNTE } r1 \text{ A } r2 \text{ COM } r1.\text{SID}=r2.\text{SID E } r1.\text{CID} \neq r2.\text{CID}$
- ▶ $r4 \leftarrow \text{PROJETE } r3 \text{ SOBRE SID}$



Exercício Resolvido 1

- Listar todos os cursos que Bob NÃO assiste

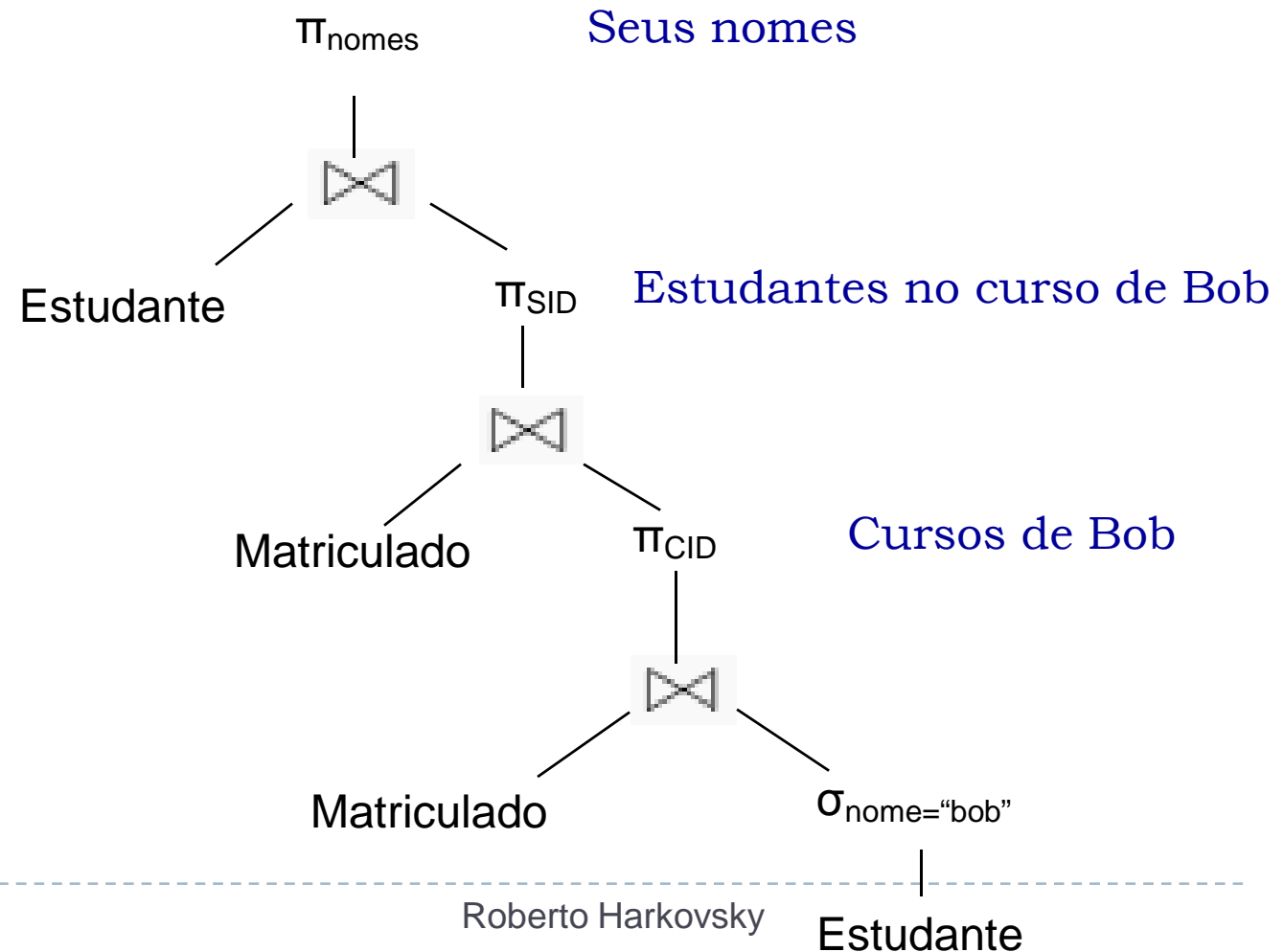


Exercício Resolvido 1

- ▶ Listar todos os cursos que Bob NÃO assiste
- ▶ $r1 \leftarrow \text{SELECIONE Estud COM nome='Bob'}$
- ▶ $r2 \leftarrow \text{JUNTE } r1 \text{ A Matric COM } r1.\text{CID} = \text{Matric.CID}$
- ▶ $r3 \leftarrow \text{PROJETE } r2 \text{ SOBRE CID}$
- ▶ $r4 \leftarrow \text{PROJETE Curso SOBRE CID}$
- ▶ $r5 \leftarrow r4 - r3$

Exercício Resolvido 2

- Listar nomes dos estudantes na classe de Bob



Exercício Resolvido 2

- ▶ Listar nomes dos estudantes na classe de Bob
- ▶ $r1 \leftarrow \text{SELECIONE Estud COM nome='Bob'}$
- ▶ $r2 \leftarrow \text{JUNTE } r1 \text{ A Matric COM } r1.CID = \text{Matric.CID}$
- ▶ $r3 \leftarrow \text{PROJETE } r2 \text{ SOBRE CID}$
- ▶ $r4 \leftarrow \text{JUNTE } r3 \text{ A Matric COM } r3.CID = \text{Matric.CID}$
- ▶ $r5 \leftarrow \text{PROJETE } r4 \text{ SOBRE CID}$
- ▶ $r6 \leftarrow \text{JUNTE } r5 \text{ A Estud COM } r5.SID = \text{Estud.SID}$
- ▶ $r7 \leftarrow \text{PROJETE } r6 \text{ SOBRE nomes}$